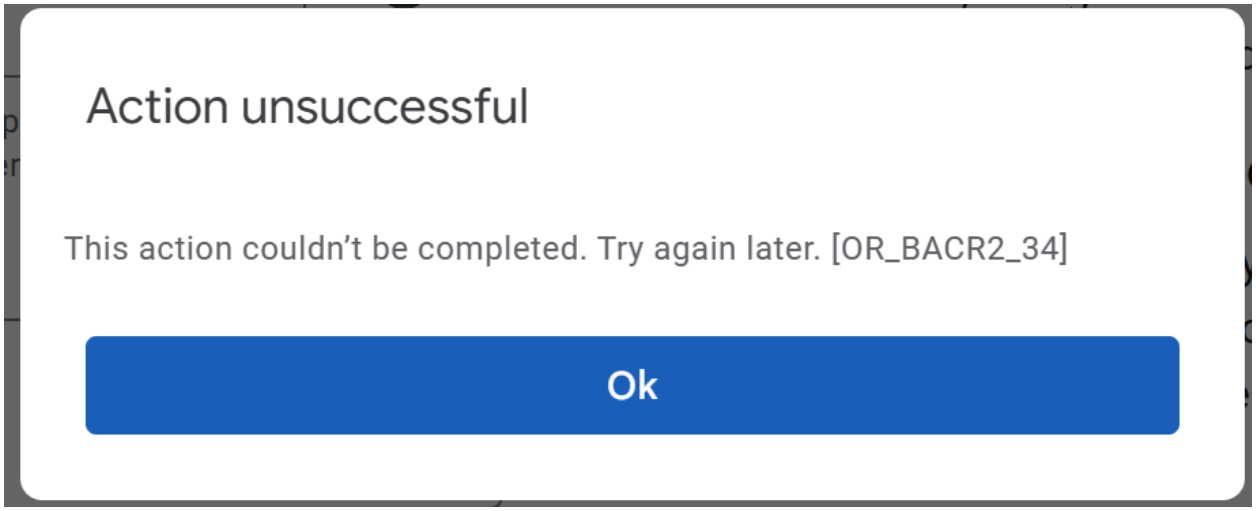
Lab 2 - Cloud Computing

Okiki Ojo (100790236)

GitHub Link: https://github.com/okikio-school/cloud-computing-labs/tree/main/Lab%202
Video Links:

- **Smart Meter Video**: SmartMeter - Screen Recording 2025-01-23 011927.mp4
- **Design Part Video**: Design Part - Screen Recording 2025-01-23 020435.mp4

Activity

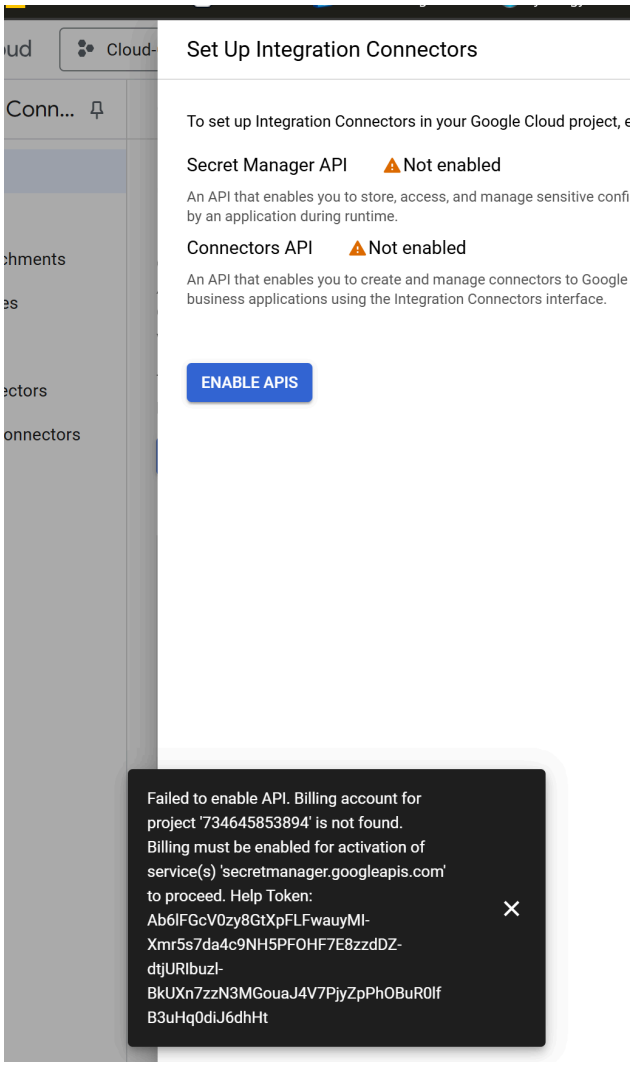


If you want to cancel all activity or usage, make a payment here to cover any remaining balance, and then visit Account management to close the billing account.

Your signup is on hold. We will email you within 48 hours to let you know if your signup is approved.

Your balance CA$0.00

To get around this issue I tried using MicroK8s locally within WSL, I was able to get Redis and MySQL both working, but Google Cloud refuses to let me setup Integration Connectors without a Billing Account and Refuses to accept my card in order to create the Billing Account.



```
okikio@Okiki-PC:~$ microk8s enable dashboard
microk8s enable dns
microk8s enable registry
Infer repository core for addon dashboard
Enabling Kubernetes Dashboard
Infer repository core for addon metrics-server
Enabling Metrics-Server
serviceaccount/metrics-server created
clusterrole.rbac.authorization.k8s.io/system:aggregated-metric
clusterrole.rbac.authorization.k8s.io/system:metrics-server cr
rolebinding.rbac.authorization.k8s.io/metrics-server-auth-read
clusterrolebinding.rbac.authorization.k8s.io/metrics-server:sy
or created
```

```
okikio@Okiki-PC:~$ microk8s kubectl get all --all-namespaces
NAMESPACE            NAME                                         READY   STATUS      RESTARTS   AGE
container-registry   pod/registry-766d4b9987-bctv8                0/1     Pending     0          11s
kube-system          pod/calico-kube-controllers-759cd8b574-r4bzg 1/1     Running     0          2m1s
kube-system          pod/calico-node-smvv9                        1/1     Running     0          2m1s
kube-system          pod/coredns-7896dbf49-q4vjk                  1/1     Running     0          2m1s
kube-system          pod/dashboard-metrics-scraper-6b96ff7878-5n5cp 1/1   Running     0          15s
kube-system          pod/hostpath-provisioner-5fbc49d86c-kktxk    1/1     Running     0          12s
kube-system          pod/hostpath-provisioner-okiki-pc-vswl4      0/1     Completed   0          7s
kube-system          pod/kubernetes-dashboard-7d869bcd96-j96tr    1/1     Running     0          15s
kube-system          pod/metrics-server-df8dbf7f5-bqdtb           0/1     Running     0          16s

NAMESPACE            NAME                                   TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)
   AGE
container-registry   service/registry                       NodePort    10.152.183.166   <none>        5000:32000/TCP
   11s
default              service/kubernetes                     ClusterIP   10.152.183.1     <none>        443/TCP
   2m7s
kube-system          service/dashboard-metrics-scraper      ClusterIP   10.152.183.94    <none>        8000/TCP
   15s
kube-system          service/kube-dns                       ClusterIP   10.152.183.10    <none>        53/UDP,53/TCP,9153/TCP
   2m5s
kube-system          service/kubernetes-dashboard           ClusterIP   10.152.183.47    <none>        443/TCP
   15s
kube-system          service/metrics-server                 ClusterIP   10.152.183.112   <none>        443/TCP
   16s

NAMESPACE     NAME                          DESIRED   CURRENT   READY   UP-TO-DATE   AVAILABLE   NODE SELECTOR            A
GE
kube-system   daemonset.apps/calico-node    1         1         1       1            1           kubernetes.io/os=linux   2
m7s

NAMESPACE            NAME                                       READY   UP-TO-DATE   AVAILABLE   AGE
container-registry   deployment.apps/registry                   0/1     1            0           11s
kube-system          deployment.apps/calico-kube-controllers    1/1     1            1           2m7s
kube-system          deployment.apps/coredns                    1/1     1            1           2m5s
kube-system          deployment.apps/dashboard-metrics-scraper  1/1     1            1           15s
kube-system          deployment.apps/hostpath-provisioner       1/1     1            1           12s
kube-system          deployment.apps/kubernetes-dashboard       1/1     1            1           15s
kube-system          deployment.apps/metrics-server             0/1     1            0           16s

NAMESPACE            NAME                                                  DESIRED   CURRENT   READY   AGE
container-registry   replicaset.apps/registry-766d4b9987                   1         1         0       11s
kube-system          replicaset.apps/calico-kube-controllers-759cd8b574    1         1         1       2m1s
kube-system          replicaset.apps/coredns-7896dbf49                     1         1         1       2m1s
kube-system          replicaset.apps/dashboard-metrics-scraper-6b96ff7878  1         1         1       15s
kube-system          replicaset.apps/hostpath-provisioner-5fbc49d86c       1         1         1       12s
kube-system          replicaset.apps/kubernetes-dashboard-7d869bcd96       1         1         1       15s
kube-system          replicaset.apps/metrics-server-df8dbf7f5              1         1         0       16s
okikio@Okiki-PC:~$ microk8s dashboard-proxy
```

```
okikio@Okiki-PC:~$ microk8s dashboard-proxy
Checking if Dashboard is running.
Infer repository core for addon dashboard
Waiting for Dashboard to come up.
Trying to get token from microk8s-dashboard-token
Waiting for secret token (attempt 0)
Dashboard will be available at https://127.0.0.1:10443
Use the following token to login:
eyJhbGciOiJSUzI1NiIsImtpZCI6ImRxSGxlcW5ZVU9lNnFaTkVjVzhlR3l6VFJvc2UwVVZsNXFETXFzUzJiTzAifQ.eyJpc3MiOiJrdWJlcm5ldGVzL3NlcnZpY2VhY2NvdW50Iiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9uYW1lc3BhY2UiOiJrdWJlLXN5c3RlbSIsImt1YmVybmV0ZXMuaW8vc2VydmljZWFjY291bnQvc2VjcmV0Lm5hbWUiOiJtaWNyb2s4cy1kYXNoYm9hcmQtdG9rZW4iLCJrdWJlcm5ldGVzLmlvL3NlcnZpY2VhY2NvdW50L3NlcnZpY2UtYWNjb3VudC5uYW1lIjoiZGVmYXVsdCIsImt1YmVybmV0ZXMuaW8vc2VydmljZWFjY291bnQvc2VydmljZS1hY2NvdW50LnVpZCI6ImIwMTBkYjgyLTlhZWYtNGIyYy1hNGFkLTZhZTdhYjFkY2E3YiIsInN1YiI6InN5c3RlbTpzZXJ2aWNlYWNjb3VudDprdWJlLXN5c3RlbTpkZWZhdWx0In0.RcJFAaw6_eJfkXaGvPm658iP5LbHWI2MMfalYiVKooH0yw7XrHSwxelTkRfVt-Zk543brW3HqG4bbVBVkT2H2q4oSR7FPc_zxDTBRWY9SxzZW-PFQl_7MN78dQvd7jUML3r5u-U4GRdp671soj8ALlVNI7ycxgybe0_uwmaJhlwk2OdaXjdK8VpVudljEYsRFdTEX3HdrFu2HW67HIQLa3OMN9_Jd6Tv09OJqbenoePEouC8nr4PRYj8AOf3ZDFO1Sv4elsVmGov0sKAJBsEVclyZR1gFPekKorYqdR-m6uk7NuvD83Iky0agv24yL9nOkbLh0aPMm15m2RUy2JjLg
^Cokikio@Okiki-PC:~$ ls
snap
okikio@Okiki-PC:~$ pwd
/home/okikio
okikio@Okiki-PC:~$ mkdir projects
okikio@Okiki-PC:~$ cd projects
okikio@Okiki-PC:~/projects$ git clone https://github.com/zubxxr/SOFE4630U-MS2.git
Cloning into 'SOFE4630U-MS2'...
remote: Enumerating objects: 705, done.
remote: Counting objects: 100% (45/45), done.
remote: Compressing objects: 100% (31/31), done.
remote: Total 705 (delta 25), reused 14 (delta 14), pack-reused 660 (from 3)
Receiving objects: 100% (705/705), 24.59 MiB | 19.67 MiB/s, done.
Resolving deltas: 100% (212/212), done.
okikio@Okiki-PC:~/projects$ cd SOFE4630U-MS2/mySQL
okikio@Okiki-PC:~/projects/SOFE4630U-MS2/mySQL$ kubectl create -f mysql-deploy.yaml
Command 'kubectl' not found, but can be installed with:
sudo snap install kubectl
okikio@Okiki-PC:~/projects/SOFE4630U-MS2/mySQL$ microk8s kubectl create -f mysql-deploy.yaml
deployment.apps/mysql-deployment created
okikio@Okiki-PC:~/projects/SOFE4630U-MS2/mySQL$ microk8s kubectl get deployment
NAME               READY   UP-TO-DATE   AVAILABLE   AGE
mysql-deployment   1/1     1            1           23s
okikio@Okiki-PC:~/projects/SOFE4630U-MS2/mySQL$ microk8s kubectl create -f mysql-service.yaml
service/mysql-service created
okikio@Okiki-PC:~/projects/SOFE4630U-MS2/mySQL$ kubectl get service
```

```
okikio@Okiki-PC:~$ pwd
/home/okikio
okikio@Okiki-PC:~$ mkdir projects
okikio@Okiki-PC:~$ cd projects
okikio@Okiki-PC:~/projects$ git clone https://github.com/zubxxr/SOFE4630U-MS2.git
Cloning into 'SOFE4630U-MS2'...
remote: Enumerating objects: 705, done.
remote: Counting objects: 100% (45/45), done.
remote: Compressing objects: 100% (31/31), done.
remote: Total 705 (delta 25), reused 14 (delta 14), pack-reused 660 (from 3)
Receiving objects: 100% (705/705), 24.59 MiB | 19.67 MiB/s, done.
Resolving deltas: 100% (212/212), done.
okikio@Okiki-PC:~/projects$ cd SOFE4630U-MS2/mySQL
okikio@Okiki-PC:~/projects/SOFE4630U-MS2/mySQL$ kubectl create -f mysql-deploy.yaml
Command 'kubectl' not found, but can be installed with:
sudo snap install kubectl
okikio@Okiki-PC:~/projects/SOFE4630U-MS2/mySQL$ microk8s kubectl create -f mysql-deploy.yaml
deployment.apps/mysql-deployment created
okikio@Okiki-PC:~/projects/SOFE4630U-MS2/mySQL$ microk8s kubectl get deployment
NAME               READY   UP-TO-DATE   AVAILABLE   AGE
mysql-deployment   1/1     1            1           23s
okikio@Okiki-PC:~/projects/SOFE4630U-MS2/mySQL$ microk8s kubectl create -f mysql-service.yaml
service/mysql-service created
okikio@Okiki-PC:~/projects/SOFE4630U-MS2/mySQL$ kubectl get service
Command 'kubectl' not found, but can be installed with:
sudo snap install kubectl
okikio@Okiki-PC:~/projects/SOFE4630U-MS2/mySQL$ microk8s kubectl get service
NAME            TYPE           CLUSTER-IP       EXTERNAL-IP   PORT(S)          AGE
kubernetes      ClusterIP      10.152.183.1     <none>        443/TCP          6m30s
mysql-service   LoadBalancer   10.152.183.138   <pending>     3306:32656/TCP   13s
okikio@Okiki-PC:~/projects/SOFE4630U-MS2/mySQL$ set alias=kubectl microk8s kubectl
okikio@Okiki-PC:~/projects/SOFE4630U-MS2/mySQL$ kubectl get service
Command 'kubectl' not found, but can be installed with:
sudo snap install kubectl
okikio@Okiki-PC:~/projects/SOFE4630U-MS2/mySQL$ unset alias=kubectl
okikio@Okiki-PC:~/projects/SOFE4630U-MS2/mySQL$ unset alias=kubectl microk8s kubectl
okikio@Okiki-PC:~/projects/SOFE4630U-MS2/mySQL$ alias kubectl="microk8s kubectl"
okikio@Okiki-PC:~/projects/SOFE4630U-MS2/mySQL$ kubectl get service
NAME            TYPE           CLUSTER-IP       EXTERNAL-IP   PORT(S)          AGE
kubernetes      ClusterIP      10.152.183.1     <none>        443/TCP          7m57s
mysql-service   LoadBalancer   10.152.183.138   <pending>     3306:32656/TCP   100s
okikio@Okiki-PC:~/projects/SOFE4630U-MS2/mySQL$ mysql -uusr -psofe4630u -h
^Cokikio@Okiki-PC:~/projects/SOFE4630U-MS2/mySQL$kubectl get serviceh
NAME            TYPE           CLUSTER-IP       EXTERNAL-IP   PORT(S)          AGE
kubernetes      ClusterIP      10.152.183.1     <none>        443/TCP          35m
```

```
okikio@Okiki-PC:~/projects/SOFE4630U-MS2/mySQL$ sudo apt install mysql-client-core-8.0
[sudo] password for okikio:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  mysql-client-core-8.0
0 upgraded, 1 newly installed, 0 to remove and 82 not upgraded.
Need to get 2727 kB of archives.
After this operation, 61.7 MB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 mysql-client-core-8.0 amd64 8.0.41-0ubuntu0.24.04.1 [2727
kB]
Fetched 2727 kB in 3s (914 kB/s)
Selecting previously unselected package mysql-client-core-8.0.
(Reading database ... 40794 files and directories currently installed.)
Preparing to unpack .../mysql-client-core-8.0_8.0.41-0ubuntu0.24.04.1_amd64.
deb ...
Unpacking mysql-client-core-8.0 (8.0.41-0ubuntu0.24.04.1) ...
Setting up mysql-client-core-8.0 (8.0.41-0ubuntu0.24.04.1) ...
Processing triggers for man-db (2.12.0-4build2) ...
okikio@Okiki-PC:~/projects/SOFE4630U-MS2/mySQL$ mysql -uusr -psofe4630u -h10.152.183.138
mysql: [Warning] Using a password on the command line interface can be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 8
Server version: 8.0.32 MySQL Community Server - GPL

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use Readings;
rType( ID int primary key, type varchar(50), cost float);
insert into meterType values(1,'boston',100.5);
insert into meterType values(2,'denver',120);
insert into meterType values(3,'losang',155);
select * from meterType where cost>=110; Database changed
mysql> create table meterType( ID int primary key, type varchar(50), cost float);
Query OK, 0 rows affected (0.02 sec)

mysql> insert into meterType values(1,'boston',100.5);
Query OK, 1 row affected (0.01 sec)

mysql> insert into meterType values(2,'denver',120);
Query OK, 1 row affected (0.00 sec)

mysql> insert into meterType values(3,'losang',155);
Query OK, 1 row affected (0.01 sec)

mysql> select * from meterType where cost>=110;
+----+--------+------+
| ID | type   | cost |
+----+--------+------+
|  2 | denver |  120 |
|  3 | losang |  155 |
+----+--------+------+
2 rows in set (0.00 sec)

mysql> exit
Bye
okikio@Okiki-PC:~/projects/SOFE4630U-MS2/mySQL$ cd ../Redis
okikio@Okiki-PC:~/projects/SOFE4630U-MS2/Redis$ kubectl create -f redis.yaml
service/redis created
deployment.apps/redis created
okikio@Okiki-PC:~/projects/SOFE4630U-MS2/Redis$ kubectl get service
NAME            TYPE           CLUSTER-IP       EXTERNAL-IP   PORT(S)          AGE
kubernetes      ClusterIP      10.152.183.1     <none>        443/TCP          37m
```

```
okikio@Okiki-PC:~/projects/SOFE4630U-MS2/mySQL$ cd ../Redis
okikio@Okiki-PC:~/projects/SOFE4630U-MS2/Redis$ kubectl create -f redis.yaml
service/redis created
deployment.apps/redis created
okikio@Okiki-PC:~/projects/SOFE4630U-MS2/Redis$ kubectl get service
NAME            TYPE           CLUSTER-IP       EXTERNAL-IP   PORT(S)          AGE
kubernetes      ClusterIP      10.152.183.1     <none>        443/TCP          37m
mysql-service   LoadBalancer   10.152.183.138   <pending>     3306:32656/TCP   30m
redis           LoadBalancer   10.152.183.202   <pending>     6379:31372/TCP   16s
okikio@Okiki-PC:~/projects/SOFE4630U-MS2/Redis$ redis-cli -h 10.152.183.202
-a sofe4630u
Command 'redis-cli' not found, but can be installed with:
sudo apt install redis-tools          # version 5:7.0.14-2, or
sudo apt install valkey-redis-compat  # version 7.2.5+dfsg1-2ubuntu4~24.04.1
okikio@Okiki-PC:~/projects/SOFE4630U-MS2/Redis$ sudo apt install redis-tools
[sudo] password for okikio:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libatomic1 libjemalloc2 liblzf1
Suggested packages:
  ruby-redis
The following NEW packages will be installed:
  libatomic1 libjemalloc2 liblzf1 redis-tools
0 upgraded, 4 newly installed, 0 to remove and 82 not upgraded.
Need to get 1440 kB of archives.
After this operation, 7422 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://archive.ubuntu.com/ubuntu noble-updates/main amd64 libatomic1 amd64 14.2.0-4ubuntu2~24.04 [10.5 kB]
Get:2 http://archive.ubuntu.com/ubuntu noble/universe amd64 libjemalloc2 amd64 5.3.0-2build1 [256 kB]
Get:3 http://archive.ubuntu.com/ubuntu noble/universe amd64 liblzf1 amd64 3.6-4 [7624 B]
Get:4 http://archive.ubuntu.com/ubuntu noble/universe amd64 redis-tools amd64 5:7.0.15-1build2 [1165 kB]
Fetched 1440 kB in 3s (570 kB/s)
Selecting previously unselected package libatomic1:amd64.
(Reading database ... 40820 files and directories currently installed.)
Preparing to unpack .../libatomic1_14.2.0-4ubuntu2~24.04_amd64.deb ...
Unpacking libatomic1:amd64 (14.2.0-4ubuntu2~24.04) ...
Selecting previously unselected package libjemalloc2:amd64.
Preparing to unpack .../libjemalloc2_5.3.0-2build1_amd64.deb ...
Unpacking libjemalloc2:amd64 (5.3.0-2build1) ...
Selecting previously unselected package liblzf1:amd64.
Preparing to unpack .../liblzf1_3.6-4_amd64.deb ...
Unpacking liblzf1:amd64 (3.6-4) ...
Selecting previously unselected package redis-tools.
Preparing to unpack .../redis-tools_5%3a7.0.15-1build2_amd64.deb ...
Unpacking redis-tools (5:7.0.15-1build2) ...
Setting up libjemalloc2:amd64 (5.3.0-2build1) ...
Setting up liblzf1:amd64 (3.6-4) ...
Setting up libatomic1:amd64 (14.2.0-4ubuntu2~24.04) ...
Setting up redis-tools (5:7.0.15-1build2) ...
Processing triggers for man-db (2.12.0-4build2) ...
Processing triggers for libc-bin (2.39-0ubuntu8.3) ...
okikio@Okiki-PC:~/projects/SOFE4630U-MS2/Redis$ redis-cli -h 10.152.183.202 -a sofe4630u
Warning: Using a password with '-a' or '-u' option on the command line interface may not be safe.
10.152.183.202:6379>  select 0
 *OK
10.152.183.202:6379> set var 100
OK
10.152.183.202:6379>  get var
"100"
10.152.183.202:6379>  keys *
1) "var"
10.152.183.202:6379> del var
(integer) 1
10.152.183.202:6379> keys *
(empty array)
10.152.183.202:6379> exit
okikio@Okiki-PC:~/projects/SOFE4630U-MS2/Redis$ cd code
okikio@Okiki-PC:~/projects/SOFE4630U-MS2/Redis/code$ code .
```

```
okikio@Okiki-PC:~/projects/SOFE4630U-MS2/Redis/code$ source ~/.bashrc
(base) okikio@Okiki-PC:~/projects/SOFE4630U-MS2/Redis/code$ pip install redis
Collecting redis
  Downloading redis-5.2.1-py3-none-any.whl.metadata (9.1 kB)
Downloading redis-5.2.1-py3-none-any.whl (261 kB)
Installing collected packages: redis
Successfully installed redis-5.2.1
(base) okikio@Okiki-PC:~/projects/SOFE4630U-MS2/Redis/code$ python SendImage
.py
Image sent
(base) okikio@Okiki-PC:~/projects/SOFE4630U-MS2/Redis/code$ python ReceiveImage.py
Image recieved, check ./recieved.jpg
(base) okikio@Okiki-PC:~/projects/SOFE4630U-MS2/Redis/code$
```

```
(base) okikio@Okiki-PC:~/projects/SOFE4630U-MS2/Redis/code$ mysql -u
usr -psofe4630u -h10.152.183.138
mysql: [Warning] Using a password on the command line interface can
be insecure.
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 9
Server version: 8.0.32 MySQL Community Server - GPL

Copyright (c) 2000, 2025, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input
statement.

mysql> use Readings;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> create table SmartMeter( ID int primary key, time bigint, pro
file_name varchar(100), temperature double, humidity double, pressur
e double);
Query OK, 0 rows affected (0.08 sec)

mysql> exit
Bye
(base) okikio@Okiki-PC:~/projects/SOFE4630U-MS2/Redis/code$
```

☐	imagesData	Google-managed	projects/clou...	—	⋮
☐	labelsCSV	Google-managed	projects/clou...	—	⋮

```
mysql> -- Create the table to store the data
mysql> CREATE TABLE CarObservation (
    ->     Timestamp BIGINT PRIMARY KEY,
    ->     Car1_Location_X FLOAT,
    ->     Car1_Location_Y INT,
r1_Location_Z FLOAT,
    Car2_Lo    ->     Car1_Location_Z FLOAT,
    ->     Car2_Location_X FLOAT,
    ->     Car2_Location_Y INT,
    ->     Car2_Location_Z FLOAT,
    ->     Occluded_Image_View VARCHAR(255),
    ->     Occluding_Car_View VARCHAR(255),
    ->     Ground_Truth_View VARCHAR(255),
    ->     PedestrianLocationX_TopLeft INT,
    ->     PedestrianLocationY_TopLeft INT,
    ->     PedestrianLocationX_BottomRight INT,
    ->     PedestrianLocationY_BottomRight INT
    -> );
Query OK, 0 rows affected (0.04 sec)

mysql>
```

Discussion:

- What's the difference between Source and Sink connectors?
  - Source = A Data source
  - Sink = Destination the data is being sent to
- What's the applications of the connectors?
  - To connect Data Source to Destinations storage systems, e.g. Pub/Sub to Redis or MySQL

Design:

We will continue using the same dataset used in the first milestone. However, we will use the Whole dataset, not only the CSV file. The dataset:

- can be accessed from https://github.com/GeorgeDaoud3/SOFE4630U-Design

- contains a folder, Dataset_Occluded_Pedestrian, of images
- contains the Labels.csv file, you used in the first milestone.

You needed to

- create two topics one for the records of the CSV file and the other for the images.
- Deploy a MySQL server and create an empty table within it to accommodate the records of the CSV file.
- Create an application integration to automatically store the records published in the topic into the MySQL database.
- Use the same script, we written in the first milestone to publish the messages into the topic.
- Deploy a Redis server to store the images.
- Create an application integration to automatically store the images published in the other topic into the Redis datastorage.
- Write a python script that will publish the images to the topic. The script should
    - Read search for all the images in the folder.
    - For each image
        - Read the image.
        - Serialize it.
        - Publish the message into the topic using the image name as the message key and the serialized image as the message value.

Deliverables

1. A GitHub link to the scripts used in the Design part.
2. A report that includes the discussion and the design parts.
3. An audible video of about 5 minutes showing the design part. The video should include proofs of the successful integration of the Cloud Pub/Sub with the connectors.

Put the GitHub link and video links inside your report, and submit the report.